

Active System Manager Integration for VMware vRealize Orchestrator Users Guide



Notes, cautions, and warnings

 **NOTE:** A NOTE indicates important information that helps you make better use of your computer.

 **CAUTION:** A CAUTION indicates either potential damage to hardware or loss of data and tells you how to avoid the problem.

 **WARNING:** A WARNING indicates a potential for property damage, personal injury, or death.

Copyright © 2015 Dell Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. Dell™ and the Dell logo are trademarks of Dell Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

2015 - 09

Rev. A00

Contents

1 Overview.....	5
2 About The Document.....	6
3 Executive Summary.....	7
4 Installing ASM vRO Integration.....	8
Installation Prerequisite.....	8
Installation Procedure.....	8
5 Validating the Installation.....	10
Validating Package Content.....	10
Validate the ASM vRO Integration API.....	10
Validating ASM vRO Integration Inventory Object.....	10
6 API for ASM vRO Integration.....	11
API Types.....	11
API Objects.....	12
ASMAppliance.....	12
ASMApplianceHostConfig.....	14
ASMHostConfigManager.....	15
Service.....	15
ServiceComponent.....	16
ServiceFolder.....	16
Template.....	16
TemplateFolder.....	16
7 Workflow and Actions.....	17
Workflow.....	17
Configuration Workflow.....	17
Sample workflow.....	17
Actions.....	19
com.dell.library.asm.....	19
com.dell.library.asm.configuration	20
8 Using vRealize Orchestrator.....	22
Adding a New ASM Appliance Object to vRO Inventory.....	22
Deploy a New Service to vRO.....	22

Add a Server to an Existing Service.....	23
Teardown the Service.....	23
9 Integrating with vRealize Automation.....	24
Prerequisite for vRA Integration.....	24
Non Interactive Workflow as vRA Service Blueprints.....	24
Interactive Workflow as vRA Service Blueprints.....	25
Workflow Output.....	26
10 Troubleshooting.....	27
vRO Exceptions.....	27
vRO Application Server Log.....	27
ASM Log.....	28
11 Uninstalling ASM vRO Integration.....	29

Overview

The Dell Active System Manager integration for vRealize Orchestrator (vRO), formerly known as vCenter Orchestrator (vCO), enables key automation functions, which can be embedded into vRealize Orchestrator workflow. The workflow assists in provisioning infrastructure, capacity scaling, remediation actions, and service teardown. These features help IT teams to broaden their business solutions, improve responsiveness, and simplify complex IT processes.

The Integration is developed using the VMware vRO SDK and Active System Manager's newly published Rest APIs. The ASM Integration for vRO will be given to ASM customers to install it into their vRealize Orchestrator environment. After installing the integration, customers can use the workflows packaged with the plugin or they can create their own workflows using the ASM attributes implemented in the creation of the plugin. IT administrators use the two main management products for this integration:

- VMware's vCenter Server
- VMware vRealize Automation Center

ASM integration for vRealize Orchestrator also enables the vRealize Automation (vRA) to integrate with VMware vRealize Orchestrator (vRO).

VMware vRealize Orchestrator (vRO) allows:

- Any template in ASM, whether infrastructure or workload, to be published as a service in vCAC portal.
- Enables any ASM function to be called from vCO for broader orchestration.

vRealize Automation (vRA, formerly known as vCloud Automation Center) enables:

- Consumption portal and service catalog
- Broad workflow and process integration
- Aggregation of **anything as a service**, including any ASM-based template or ASM action such as scaling and remediation

This integration enables unique services not supported by VMware, physical, hybrid physical-virtual, multi-hypervisor, infrastructure scaling, remediation and so on. The ASM vRO Integration supports vCO and vRO versions 5.5 and 6.0.

About The Document

This documentation provides information regarding:

- Installing ASM vRO intergration
- Validating the installation
- Configuring ASM vRO API
- Workflow and Action
- Using vRealize Orchestrator
- Integrating with vRealize Automation
- Troubleshooting
- Uninstalling ASM vRO Integration

Executive Summary

ASM is a service-based management console that automates end-to-end template based provisioning and lifecycle management of infrastructure and workloads in today's modern data center. ASM and its REST base API, enables end-to-end integration and automation with additional tools and management consoles in the data center. Active System Manager integration of VMware vRealize Orchestrator, automates customers cloud and integrate VMware management systems with Active System Manager. Workflow packages are developed using the Orchestrator to achieve step-by-step process automation for greater flexibility and automation across physical and virtual infrastructure and multiple management products.

Installing ASM vRO Integration

Before installing ASM vRO integration, ensure that you have met the installation prerequisite:

Installation Prerequisite

To install ASM vRO Integration, vRO environment must meet the following criteria before installing the Integration:

- The vRO server must be able to communicate to the ASM appliance server(s) through HTTPS protocol (port 443).
- The vRO server must be in a clean state. Either the ASM vRO Integration is never been installed before or it should be removed completely. To ensure that the environment is clean, see the chapter – **Uninstalling ASM vRO Integration**.
- The vRO configuration manager server must be running.
- The workstation must be able to reach the vRO configuration server, and the `asmvropugin.vmoapp` must be downloaded to the workstation.

 **NOTE:** To avoid data loss, make sure there is no workflow running in the vRO during the installation procedure, as restart is required.

Installation Procedure

To install ASM vRO Integration:

1. Open vRO Configuration Manager on the browser. Usually, it is "`https://<VRO_HOSTNAME>/VCO-config/`" or "`https:// <vRA_HOSTNAME>/vco-config/`".
2. In VMware vRealize Orchestrator window, login as a user with administrative privileges. Click the **Plug-ins** option in the left pane of the **VMware vRealize Orchestrator configuration** page.
3. After you click the **Plug-ins** option, you will get all the list of existing integrations. Scroll down the Plug-ins page on the right panel to see **Install new plug-in** section at the bottom of the page.
4. Click the **Search** button beside **Plug-in file** option, browse and select the **vmoapp file**. Click **Open**. To install the vmoapp file, click the **Upload and install** button. You will be prompted to accept the licence. Read the displayed information and click **I accept the term of the License agreement**. It will direct you to Plug-ins page.
5. You can see a text in the Plug-ins page says, **Dell Active System Manager (1.0.0 build 1) New plug-in installed**.
6. Restart the vRO application server. To do so you can either establish `ssh` to the vRO appliance server as the root user and run the command, `/etc/init.d/vco-server restart`, or you can restart the vRO application server through the vRO configuration manager. To restart vRO application server through the vRO configuration manager, click **Startup** Options in the left pane, Server startup options window appears. Click **Restart service** under the option or click **Restart the vRO Configuration Server** under **vRO Configuration Server**. This completes the installation.

7. After restarting vRO server, it returns to the main menu page, click **vRealize Orchestrator Client**.
8. Enter your administrator credential to log in.
9. After you log in, you can see ASM plugin on Home page.

Validating the Installation

To validate the installation, make sure:

- The package contents are imported.
- ASM API is available.
- New Inventory object is created.

Validating Package Content

1. When the ASM vRO Integration is successfully installed, new contents should appear in vRO. To inspect these, launch the vRO client, and select the Design view.
2. You will see six tabs in the left panel to inspect different types of vRO contents. First select the Packages tab. It shows the list of existing packages and additional information about each package.
3. Locate the package, com.dell.plugin.asm. Now click the **Workflow** tab in the right pane of the vRO client.
4. Click the **Actions** tab in the right panel, you can see action elements are listed.
5. Finally to check whether the ASM API is associated with these elements or not, click the **Used Plugins** tab in the right pane. It should list the ASM API with its name and version information.

Validate the ASM vRO Integration API

Make sure the ASM Integration API is available. One quick way to validate this is by finding the ASM API in the API Explorer of the vRO client.

1. In the top right corner of the vRO client, click on the **Tools** tab and expand it.
2. Then click **API Explorer selection** from the drop-down menu.
It brings up a window that shows the API Explorer, locate **ASM SDK** module from the explorer.

Validating ASM vRO Integration Inventory Object

Make sure a new Inventory object is created for the ASM vRO Integration. Click the **Inventory** tab from the left pane to confirm that there is an inventory object named, **Dell Active System Manager**. The object can be expanded, but it should be empty at this point.

API for ASM vRO Integration

The ASM vRO Integration includes a package with useful elements such as the actions, workflows, and inventory. The ASM vRO Integration is operated through the ASM plugin API. The API includes the SDK module, types, objects, and their methods. The types are used for representing the API objects into the inventory or finder view. The API objects contain attributes and methods that are useful for automating procedures which require ASM operations. For more information, see the chapter-**Workflows and Actions**. You can also see the description of the types, objects, attributes, and methods from the API Explorer.

API Types

The API types provide mapping of the API objects to the finder objects. In scripting, API types are also used to define the type of required input, output, and attribute.

ASM — ASM PLUG IN:

- Scripting object: FinderResult
- An abstract type, FinderResult, is a top-level vRO type that is used to define an object as an inventory. This type is used for the root level inventory object of the ASM vRO Integration.

ASM — Appliance:

- Scripting object: ASMAppliance
- This type is mapped to ASMAppliance, which is responsible for storing ASM appliance configurations and making REST API calls. The FinderResult object of this type is a child object of the ASM:ASM_PLUGIN type object. It also contains the child objects of the ASM:ServiceFolder type and the ASM:TemplateFolder type. See ASMAppliance for detail.

ASM — Service:

- Scripting object: Service
- This type is mapped to Service class. This object is a child object of the ASM:ServiceFolder type object in the finder. It also contains child objects of the ASM:ServiceComponent type. See Service for detail.

ASM — ServiceComponent:

- Scripting object: ServiceComponent
- This type is mapped to ServiceComponent. The objects of this type has the ASM:Service type object as the parent in finder. See ServiceComponent for detail.

ASM — ServiceFolder:

- Scripting object: ServiceFolder

- This type is mapped to ServiceFolder. The type and class both provide nothing much other than the inventory structure and hierarchy. The object of this type has the ASM:Appliance type object as its parent while having child objects of the ASM:Service type objects. See ServiceFolder for detail.

ASM — Template:

- Scripting object: Template
- This type is mapped to Template. In finder view, the objects of this type is structured under the ASM:TemplateFolder type object. See Template for additional information.

ASM — TemplateFolder:

- Scripting object: TemplateFolder
- This type is mapped to TemplateFolder. The type and class both provide nothing much other than the inventory structure and hierarchy. The inventory object of this type has the ASM:Appliance type object as its parent while its child objects are of the ASM:Template type. See TemplateFolder for detail.

API Objects

There are eight API objects that are included in the ASM SDK module. This section covers each object and the associated attributes and methods.

ASMAppliance

This object establishes HTTPS connection to the ASM REST webservice in order to make the requests. This is the most crucial object for the Integration.

Attributes:

- id (java.util.UUID): Identifier issued by the vRO Integration, which is used to track the ASM Appliance for internal operations
- name (String): User-given name of the ASM appliance
- serverUrl (String): URL of the ASM Appliance
- sslTrustAllCerts (boolean): Indicates if the SSL trust is enabled or disabled.

Methods:

- **addDuplicateServer:** This method clones server(s) from the configuration of an existing server component. It duplicates servers which have the same relations to other resources (For example: Duplicated servers will be added to the cluster where original server belongs to). Server pool must be able to allocate physical server resources for the chosen configuration.
 - deployment – Target Service where new server component(s) are added.
 - targetServer – ServiceComponent of server type that provides the configuration for the duplicate server(s).
 - newServerName – Name of the new server(s); if more than one instance of servers are requested, sequence number is appended to the name.
 - numInstances – Specifies the number of server instance(s) that will be added to the Service.
- **addDuplicateStorage:** This method clones storage(s) from the configuration of an existing storage component. Duplicated storages have the same relations to other resources (For example: Duplicated

storages will be used by the servers that the original storage support). Target volume must be able to allocate resources for the chosen configuration.

- deployment – Target Service where new server component(s) are added.
- targetStorage – ServiceComponent of STORAGE type that provides the configuration for the duplicate storage(s).
- newStorageName – Name of the new storage(s); if more than one instance of storages are requested, sequence number is appended to the name.
- numInstances – Specifies the number of storage instance(s) that will be added to the Service.
- **authenticate**: This method must be called first to create a session before starting other method(s).
- **deployTemplate**: Deploy a new Service with the configuration of the given Template.
 -  **NOTE**: The target Template in ASM must auto-generate the required fields (For example: the host name of the servers) other than the deployment name.
 - template – Target Template that is to provide the configuration for the new Service being deployed.
 - deploymentName – Name of the new Service.
 - description – Optional parameter that describes the deployed Service
- **getHostConfig**: Returns a ASMApplianceHostConfig object that contains the information regarding the ASM Appliance (e.g., name, serverURL, credentials, etc).
- **getService**: Retrieves the Service of the given name.
 - name – Name of the Service that should be retrieved.
- **getServiceStatus**: Returns the status of the target Service (e.g., IN_PROGRESS, COMPLETE, ERROR).
 - vsoDeployment – Service that is to be checked for its status.
- **getServiceTemplateComponents**: Retrieves a list of ServiceComponent objects from the given Service.
 - deployment – The Service to retrieve the ServiceComponent objects.
- **getServiceTemplateComponentsByType**: Retrieves a list of ServiceComponent objects, but the result is filtered by the selected resource types.
 - deployment – The Service to retrieve the ServiceComponent objects.
 - storage – Boolean value indicating that the returned list of ServiceComponents should include STORAGE type components if available.
 - server – Boolean value indicating that the returned list of ServiceComponents should include SERVER type components if available.
 - cluster – Boolean value indicating that the returned list of ServiceComponents should include CLUSTER type components if available.
 - vm – Boolean value indicating that the returned list of ServiceComponents should include VIRTUALMACHINE type components if available.
 - app – Boolean value indicating that the returned list of ServiceComponents should include SERVICE (application) type components if available.

- **getServices:** Returns a list of all the Services in the ASM server.
- **getTemplates:** Returns a list of all the published Templates in the ASM server.
- **scaleDownService:** This method removes the resources of the selected types from the chosen Service. If a resource type is selected and no resources of that type exists, it will simply the operation of the scale down.
 - deployment – The Service where the resources of the selected types are to be removed.
 - storage – Boolean value indicating that STORAGE type components are to be removed.
 - server – Boolean value indicating that SERVER type components are to be removed.
 - cluster – Boolean value indicating that CLUSTER type components are to be removed.
 - vm – Boolean value indicating that VIRTUALMACHINE type components are to be removed.
 - app – Boolean value indicating that SERVICE (application) type components are to be removed.
- **scaleDownServiceComponent:** Removes the target ServiceComponent object from the Service and returns its resources to the resource pool.
 - deployment – The Service to be down scaled.
 - component – The ServiceComponent object to be removed from the Service.
- **serviceExists:** Checks if the Service of the given name exists or not. This is useful for workflow that wait until the service is deleted.
- **tearDownService:** This method deletes the Service and returns all the resources to the resource pools.
 - ✍ **NOTE:** There is no safeguard against shared resources. Therefore, it is important to check if any of the service components is shared (For example: There is a Service that defines a cluster and supporting servers and storages). There is another Service that relies on this cluster to provision VMs if a teardown is requested for the Service.
- **tearDownServiceSelective:** This method deletes the Service and returns the selected resources to the resource pools. This method can be helpful when there are shared resources across Services.
 - deployment – Target Service to be removed; its resources are returned to the resource pools only if they are of the selected types
 - storage – Boolean value indicating to remove resources of the STORAGE type.
 - server – Boolean value indicating to remove resources of the SERVER type.
 - cluster – Boolean value indicating to remove resources of the CLUSTER type.
 - vm – Boolean value indicating to remove resources of the VIRTUALMACHINE type.
 - app – Boolean value indicating to remove resources of the SERVICE (application) type.

ASMApplianceHostConfig

This object is used mainly for persisting the ASM Appliance data in the vRO inventory. The password is not exposed, although it is stored. Using the information in this object, new ASMAppliance object can be constructed or looked up from cache depending on the case. Persisted objects are stored as vRO resources (with the password segment encrypted). In general, this class does not provide values to the regular vRO users. However, it is more important for the vRO Integration actions and the vRO administrators.

Attributes:

- domain (String): The domain of the credential provided (i.e., username and password).
- id (String): identifier issued by the vRO Integration, which is used to track the ASM Appliance for internal operations.
- name (String): User-given name of the ASM appliance.
- serverUrl (String): URL of the ASM Appliance.
- sslTrustAllCerts (boolean): Indicate if SSL trust is enabled.
- username (String): User name for the authentication.

ASMHostConfigManager

This is the class that manages the ASMApplianceHostConfig objects. It has methods to create the persisted inventory object for the ASMApplianceHostConfig objects and to remove them.

Method:

- **Create:** This method creates (persists) a new vRO resource with the given ASMApplianceHostConfig object.
 - hostConfig – ASMApplianceHostConfig object that contains required information of the ASM appliance.
- **Destroy:** This method removes the vRO resource that contains the ASMAppliance configuration detail. The same effect is achieved when the vRO administrator removes the vRO resource of the given applianceId.
 - applianceId – ID of the ASMAppliance object.

Service

The Service object describes the ASM Service with minimum essential information.

Attributes:

- deployedBy – Shows the ASM user that deployed the Service.
- deploymentDate – Shows when the Service was deployed.
- description – Shows the description of the Service if available.
- id – Service identifier given by the ASM server.
- name – Name of the Service.
- status – Status of the Service (values: IN_PROGRESS, COMPLETE, PENDING, ERROR, and CANCELLED).
- templateName - Shows the name of the Template that provided the initial configuration for the Service deployment.

ServiceComponent

The ServiceComponent object contains minimum essential data of the resources or components in Services. Resources that are described can be servers, storages, cluster, virtual machines, and applications.

Attributes:

- id (String): The ServiceComponent identifier given by the ASM server.
- ip (String): The IP address of the resource.
- name (String): Name of the ServiceComponent.
- resourceName (String): Name of the hardware resource.
- serviceTag (String): Shows the service tag.
- type (String): The ServiceComponent type – value can be SERVER, STORAGE, CLUSTER, VIRTUALMACHINE, or SERVICE (which are the application on servers or VMs).

ServiceFolder

This is a simple object used by the ASM vRO Integration for the maintenance of the inventory structure and hierarchy.

Attributes:

- Name (String): Name is displayed in the inventory or finder. It is displayed as **Services** by default.

Template

This object describes the ASM Template with essential data about the Template.

Attributes:

- category (String): The category that groups the published Templates.
- description (String): Description of the Template.
- id (String): The identifier of the Template given by the ASM server.
- name (String): Name of the Template.

TemplateFolder

This is a simple object used by the ASM vRO Integration to maintain the inventory structure and hierarchy.

Attributes :

- name (String): Name is displayed in the inventory or finder. It is displayed as **Templates** by default.

Workflow and Actions

The Dell ASM vRO Integration is prepackaged with a set of sample workflows and actions. These can be used as out-of-box to immediately enable common use cases, and it can be also used to serve as examples that can be extended by advanced users.

Workflow

The prepackaged ASM vRO Integration workflow is divided into two types:

- **Configuration workflow** – These can be used to store and remove ASM appliance configurations within vRO. Appliances can be viewed in the Inventory panel once it is stored.
- **Sample workflow** – These samples illustrate the various capabilities of the integration. They are designed to allow users to immediately begin testing the integration between vRO and their ASM appliance.

Configuration Workflow

Add ASM appliance:

This workflow asks the user to provide various configuration parameters for the ASM appliance and subsequently persists the configuration as a vRO resource. Once this workflow has been executed successfully, the appliance appears in the inventory and can be selected as an input for any of the other sample workflows.

Remove ASM appliance:

This workflow removes the selected appliance from the inventory.

Sample workflow

The sample workflow is divided into three categories:

- **Asynchronous** – This workflow is designed to make API calls ASM appliance without waiting for the completion of the corresponding operations.
- **Non-interactive** – The workflow is designed without any user inputs, and serve as examples for vRA integration or other automation purposes.
- **Synchronous** – The workflow make use of their asynchronous counterparts but include polling functionality to defer completion until the corresponding operation either succeeds or fails on the ASM appliance.

Asynchronous

Deploy service

This workflow allows users to deploy a new Service based on the template configuration chosen by the user. This workflow is completed after making the REST call to the ASM Appliance. The Service status is not tracked.

Scaledown service

Following are the set of scaledown operation workflow:

- **Scaledown Service:** Scale down the Service by removing the selected ServiceComponent (prompted during the workflow run).
- **Scaledown Service – App:** Scale down the target service by removing all resources of app type.
- **Scaledown Service – Cluster:** Scale down the target service by removing all resources of cluster type.
- **Scaledown Service – VM:** Scale down the target service by removing all resources of VM type.
- **Scaledown Service – Server:** Scale down the target service by removing all resources of server type.
- **Scaledown Service – Storage:** Scale down the target service by removing all resources of storage type.

Scaleup Service

Following are the set of sample asynchronous scaleup service workflow:

- **Scaleup Service – Server:** Scale up the service by adding more server components/resources.
- **Scaleup Service – Storage:** Scale up the service by adding more storage components/resources.

Teardown Service

Following are the set of sample asynchronous teardown service workflow:

- **Teardown service – All:** Delete the service selected (prompted during the workflow run) and return all resources to the resource pool.
- **Teardown service – Selective:** Removes the service and all of its resources selected by their type. This workflow can be used to leave resources which are utilized by other services.

Non-interactive

There are two sample non-interactive workflow. These serve as templates and can be modified by the user according to their specific environments and requirements:

- **Deploy service** - Deploy a new service based on predefined parameters that are required. In order to use this workflow, the attributes must be set as appliance, ServiceTemplate, and baseDeploymentName.
- **Scaleup service – Server**- Scaleup a service based on the predefined parameters. The attributes must be set before this workflow can be used as service, targetServiceComponent, newComponentName, numberOfServers, and asm.

To set the attributes, select the workflow and click the **Edit** button (keyboard shortcut: Ctrl + E for computer and Command + E for Apple Mac Computer). In the editing mode, select the **General** tab. You can set the values for the attributes and save it.

Synchronous

Deploy service

This workflow allows users to deploy a new service based on the template configuration chosen by the user. This workflow tracks the status of the new Service, and it completes its run when the service status is no longer displaying as **IN_PROGRESS**.

Scaledown service

This workflow scales down the service by removing the selected Service Component (prompted during the workflow run). This workflow tracks the status of the target Service, and it completes its run when the service status is no longer in **IN_PROGRESS** status.

Scaleup service

There are two sample synchronous scaleup service workflow:

- **Scaleup service – Server:** Scale up the service by adding more server components or resources. This workflow tracks the status of the target Service, and it completes its run when the Service status is no longer in **IN_PROGRESS**.
- **Scaleup service – Storage:** Scale up the service by adding more storage components or resources. This workflow tracks the status of the target Service, and it completes its run when the Service status is no longer in **IN_PROGRESS**.

Teardown service

There are two sample synchronous teardown service workflow:

- **Teardown service – All:** Delete the service selected (prompted during the workflow run) and remove all of its resources. This workflow tracks the existence of the target Service, and it completes its run when the Service no longer exists.
- **Teardown service – Selectively:** Removes the Service and all of its resources selected by their type. This workflow tracks the existence of the target Service, and it completes its run when the Service no longer exists.

Actions

The actions provide elements that can be used to build workflow by mapping the ASM vRO Integration API methods. It also provides actions that help configuring the vRO environment for managing the inventory objects. There are two groups of the actions:

- com.dell.library.asm
- com.dell.library.asm.configuration

com.dell.library.asm

Actions in com.dell.library.asm

These actions helps in mapping of the API methods from the ASMAppliance. It is responsible for making REST to sync with the ASM server:

addDuplicateServer

This is the action that calls the addDuplicateServer method of the ASMAppliance API object. It returns a service object that has the updated information.

The addDuplicateServer(s) method cannot scale up the server(s) unless the target server for duplication is configured in this way:

- The server must be created with hardware settings that specify the server source to be server pool instead of manual entry.
- The duplicated server(s) must be provisioned from the same server pool from where the original server was pulled out.

addDuplicateStorage

The method, addDuplicateStorage, is called from the ASMAppliance API object. It returns a Service object that has updated information.

DeployService

This action calls the deployTemplate method of the ASMAppliance API object. It returns a Service object that has been deployed.

getServices

This calls the getServices method from the ASMAppliance. An array of Service objects is returned.

getTemplates

This calls the getTemplates method from the ASMAppliance. An array of Template objects is returned.

scaleDownService

This action calls the scaleDownService method of the ASMAppliance. A Service object with updated information is returned.

scaleDownServiceComponent

This action calls the scaleDownServiceComponent method from the ASMAppliance. A Service object with updated information is returned.

serviceExists

This calls serviceExist method to check if the Service exists in the ASM appliance. A boolean value is returned to indicate whether it exists or not.

teardownService

This action calls teardownService method, and the target Service is returned.

teardownServiceSelective

This calls the teardownServiceSelective method from the ASMAppliance, and the target Service is returned.

com.dell.library.asm.configuration

Actions in com.dell.library.asm.configuration

The actions in this group are used for creating inventory items for the ASM appliance resources and other general helps:

addAsmApplianceConfig

This is an action that persists the given configuration of the ASMAppliance object by calling the create method of the ASMHostConfigManager.

removeAsmApplianceConfig

This removes the vRO resource used for persisting the ASMAppliance configuration (which is added through addAsmApplianceConfig action or ASMHostConfigManager.create method).

generateUniqueServiceName

This action takes a base name of Service and appends sequence number at the end to ensure the uniqueness of the given name. In order to get the list of all the Service names, it calls the getServices method from the ASMAppliance.

getDefaultAsmAppliance

This action is useful in a vRO environment that manages multiple instances of ASMAppliance objects. It picks the first persisted ASMAppliance object from the list of all the ASMAppliance resources . The action can be modified to pick a specific ASMAppliance by its ID or name.

Using vRealize Orchestrator

By using vRO with ASM integration, you can perform the following actions:

- Add a new ASM Appliance object to the vRO inventory
- Deploy a new service to vRA
- Add a server to an existing service
- Teardown a service with selected resource types

Adding a New ASM Appliance Object to vRO Inventory

For using ASM vRO plugin, the best way is to utilize the vRO workflows which are packaged with the plugin. The following instruction explain how to add an ASM Appliance object to the vRO inventory:

1. Add ASM Appliance workflow from the **Configuration Workflows** folder on the left pane, then click the **Start workflow** button on the right pane.
2. Enter the values for the name and the server URL under Common Parameter. Either enable or disable the setting to trust all SSL certificates from the server.
3. Click **Next**, and set the values for the authentication domain (ASMLOCAL by default), username and password.
4. Click **Submit**.
The workflow should run successfully. To validate whether the configuration has been added or not, go to the Resources view, expand the folders and locate the new resource item.
5. Select the **Viewer** tab to cross check the values that were used as input parameters for the workflow. The password is encrypted.

At this point, the user should be able to see the inventory objects for this ASM appliance server. Any templates or services already present in the ASM environment are also displayed.

Deploy a New Service to vRO

After adding the ASMAppliance object as a vRO resource, you can start running the workflow to deploy and update Services. In this example, the following steps describe how to deploy a new Service using vRO.

1. In run mode, click **Workflows view**, navigate **Library** → **Dell Active System Manager** → **Sample Workflow** → **Synchronous** → **Deploy Service** → **Start Workflow** (which allow vRO workflows to call ASM for infrastructure).



NOTE: The Synchronous workflow is used for demonstration.

The default ASMAppliance object should already be set as the input parameter

2. Click **Submit** to proceed.
3. Click the selection box. A message stating **Not set** is displayed. Select the template to deploy in the window that is displayed.
4. Click **Submit**.
5. Provide the service name and modify the description of the service.
6. Click **Submit**.

The workflow uses the ASM API to call the target ASM server. At this point, the workflow enters the waiting phase until the service is fully deployed.

The workflow is completed after the Service deployment is completed (For example: The status of the service is no longer **IN_PROGRESS**).

 **NOTE:** Once the cloud infrastructure is deployed, VMware vRealize Automation (vRA) can be used to grant end users self-service access to deploy virtual machines in the environment.

 **NOTE:** The deployment can be confirmed using the ASM UI or the vRO Inventory view. The Service can be viewed in the ASM UI console. The vRO Inventory view also shows the service object and its information.

Add a Server to an Existing Service

To add additional resources through the workflow, follow these steps:

1. Navigate **Library** → **Dell Active System Manager** → **Sample Workflow** → **Synchronous** → **Scaleup Service – Server Workflow**.

 **NOTE:** The Synchronous workflow is used for demonstration.

2. Set the default ASMAppliance object if the object is already a resource.
3. Click **Submit** and proceed to the following page and select the target Service.
4. Click **Submit**, and select the target server Service Component to clone and add to the Service.
5. Set the name and the number of instances (must be an integer and not a fractional value).
6. Click **Submit** to request the scaleup to the ASM server.
7. Confirm if the new resource is added to the service after completion.

Teardown the Service

1. Locate the **Teardown Service** workflow .

 **NOTE:** The synchronous workflow is used for demonstration.

 **NOTE:** The ASMAppliance object should be already set to the default value.

2. Continue and select the target Service.
3. The workflow enters the waiting phase until the service is removed.
4. Confirm that the service is removed through the ASM UI or by vRO Inventory.

Integrating with vRealize Automation

The Dell Active System Manager integration with vRealize Orchestrator and its workflows are embedded to publish vRealize Automation Service Blueprints. The result allows you to call ASM using the vRealize Orchestrator workflow which is used through the vRA self-service catalog items.

Prerequisite for vRA Integration

Make sure that you have met the following prerequisites for vRA Integration:

1. There are two distinctive modes of operations in vRA. The system administrator configures the general system settings such as tenants management, branding, email servers configuration and so on. The system administrator's account cannot be used to manage self-service items for end users. Each tenant manages its catalogs and services for vRA catalog management. Therefore, there must be tenant organization(s) in the target vRA environment other than the `vsphere.local` tenant.
2. The vRA is shipped with vRO running on the same appliance. By default, vRA is configured to use this vRO. However, if the administrator chooses to use a different vRO server, the vRA system setting must be modified to use the external vRO server where the ASM vRO Integration exists.
3. Finally, there has to be an account with sufficient privilege to add and modify the tenant service catalog items. Either the Tenant Administrator or the Service Architect role grants this privilege. To run the following workflow, either of these accounts must be in use.

Non Interactive Workflow as vRA Service Blueprints

By using the non-interactive workflow, you can create and publish vRA Service Blueprints in the easiest way. Modify the non-interactive sample workflow in vRO by setting the attributes or create a workflow with all the attributes set (no input exists to the workflow). For better illustration, assume that a sample workflow has been modified. Navigation path for the workflow is **Deploy Service workflow** → **Dell Active System Manager** → **Sample Workflow** → **Non-interactive folder**. vRA's attributes are set to deploy an ASM Template into an ASM Service to provision virtual machines in a vCenter cluster. The following instruction describes how to convert the ASM sample workflow (non-reactive) to vRA-friendly workflow:

1. Log into the vRA tenant console (It is accessible at `https://VRA_HOSTNAME/vcac/org/TENANT_NAME`) using the privileged account.
 -  **NOTE:** This step is optional. Existing Service can be used instead of using a new Service to contain the service blueprints.
2. Click the **Administration** tab.
3. Select **Catalog Management**, and then the **Services**.
4. Create a service, and add the service to an entitlement.
5. Click the **Advanced Services** tab.
6. Select the service blueprints on the left pane.

7. Click the **Add** button on the right pane to add a new service blueprint.
8. First select the **Deploy Service** workflow that was modified earlier in vRO.
9. Click the **Next** button.
10. Edit the details as needed, then proceed.
 -  **NOTE:** A blueprint form is not required for a non-interactive workflow.
11. Click **Next**, and make sure no resource is provisioned after running the workflow.
12. Click **Add** button to complete creating the service blueprint.

A new service blueprint should appear on the right pane with its given name, description, and the current status.
13. Select this item, and click the **Publish** button.

The service blueprint is now visible as a catalog item.

 -  **NOTE:** It still has to be added to an entitled Service else, the catalog item has to be added to an entitlement individually.
14. Select the **Administration** tab and click **Catalog Management**, then click the Catalog Items from the left pane.
 -  **NOTE:** The Service Blueprint is available here for additional configuration. At this point, its source is set to **Advanced Designer Service** where it was published.
15. Select the catalog item, and click **Configure**.
16. When the **Configure Catalog Item** page loads edit the description and upload a custom icon (optional), and select the Service to put the catalog item in.

Now the catalog item can be used by the end users but it must have the correct entitlement settings. To make sure that the end users have access to the new self-service catalog item, check if an entitlement exists for the users or the user group. This entitlement must include the service that contains the service blueprint or the catalog item individually.

-  **NOTE:** When an end user requests the service from this kind of catalog item, only the description of the request is required. The attributes on the workflow provide the rest of the information which is required to call ASM.

Interactive Workflow as vRA Service Blueprints

To embed interactive workflows in vRA Service Blueprints, all required data must be processed as input parameters. It means the most of the workflow bundled with the integration cannot be populated to vRA. To populate the workflow to vRA, workflow must be modified. Consider the ASM configuration workflow and add ASM Appliance. This workflow can be encapsulated as a vRA Service Blueprint without any issue as all required data are passed as input parameters. Most sample workflow contain user interaction in their schema. This will cause the workflow to pause and wait when it is called through vRA. As example, deploy Service (Asynchronous) workflow is chosen. The following instruction describes how to convert the ASM sample workflow (Interactive workflow) to vRA-friendly workflow:

1. Log in to vRO with privileged credential.
2. Duplicate the sample workflow into a folder where the vRA tenant has access.
3. Edit the duplicated workflow.
4. Remove all the attributes from the workflow.
5. Create new input parameters for **ServiceTemplate** (ASM:Template type), **DeploymentName** (string), and **DeploymentDescription** (string). There should be four input parameters in total.
6. Select the **Schema** tab, and remove all the design elements except for the action element, **DeployService**.

7. Edit the **DeployService** element. Bind the local parameters and source parameters accordingly.
8. Save the workflow and close the editor mode.
 -  **NOTE:** This workflow is ready to be encapsulated into a vRA Service Blueprint.
9. Go back to the vRA tenant console, create a new service blueprint by choosing the modified workflow. Proceed until the new Service Blueprint is added.
 -  **NOTE:** The blueprint form has auto populated fields. Updating the fields is not mandatory.
10. Publish the Service Blueprint, and grant access to the end users to this catalog item.
11. The input parameters are required to grant access to end user to use the catalog items. It can only be chosen from the vRO inventory for any ASM vRO Integration API objects, (for more information, see the **Add ASM Appliance workflow and vRO inventory**).
12. As the User Interaction elements are removed from the workflow, the vRA end user has to carefully select the **ASM:Template** object that is in the selected **ASM:Appliance** object.
 -  **NOTE:** Mismatching in selection will make the ASM server to refuse the deployment.
13. Provide the service name and description to complete all the fields (Optional).
14. Click **Submit**.

The vRA triggers the vRO workflow to call the ASM server for a new service deployment.

Workflow Output

The ASM vRO sample workflow may have output parameter with custom API object type. If the vRA needs to use the information in the output parameter, it cannot be parsed easily in the vRA layer. Therefore, the workflow should be modified to parse the parameter into primitive data type objects before running it (For example: Add a script action that takes the custom API object and outputs string objects). In this way the output data can be easily used by vRA.

Troubleshooting

You can troubleshoot ASM vRO Integration issue using:

- vRO Exceptions
- vRO Application Server Log
- ASM Log

vRO Exceptions

When an API method is invoked through actions or scripts, the operation is requested by the ASM server. However, the ASM server may return error message due to internal server issues or bad user inputs. In such cases, the message is mapped to an exception to be thrown. Typically these messages are reported with the following format:

Message<sequence number>

- Message code
- Severity
- Category
- Display message
- AgentId
- Timestamp
- Sequence number

There may be multiple messages in some cases. Plugin itself throws exceptions with appropriate messages to indicate any problem during the operation.

vRO Application Server Log

There are exceptional cases where the exception returned to the vRO which does not include enough details. In this case, the best way to track down the issue is by checking the vRO application server logs. You need access to the vRO appliance shell and read privileges for the following files (different in custom installations):

- `/var/log/vco/app/catalina.sh`
- `/var/log/vco/app/server.out`

ASM Log

If there is no obvious issue in the vRO, the last resort is to check the ASM appliance log itself.

Refer to the ASM documentations for further troubleshooting.

Go to <http://www.dell.com/asmdocs> for additional supporting documents on ASM.

For more information about ASM including how to videos, white papers, blogs, and support forum, see Dell Active System Manager page on Dell TechCenter:

<http://www.dell.com/asmtechcenter>

Uninstalling ASM vRO Integration

Before uninstalling the ASM vRO integration, make sure the integration is already installed. The **Validate the Installation** section describes the procedure of validating ASM vRO integration. If the integration is not installed, the inventory tab does not show the root FinderResult object. Additionally, you can check the API Explorer to see whether the ASM SDK module exists or not. The following steps describe how to uninstall the ASM vRO Integration:

1. Make sure no workflow is running. Ensure that you do not use the vRO server while uninstalling the vRO integration.
2. Remove the resources that persist the ASMAppliance objects. If it is not removed, it may be used again later when the ASM vRO integration is reinstalled. To remove it, you can either run the workflow, remove ASM Appliance or you can remove the resources directly. To remove the resources using the workflow, start the workflow and select the ASM Appliance to remove. Another way to remove the resources is, go to the Resources view. Select Design mode, click on **Resources tab**, then expand the folders to locate the resources that define the ASMAppliance objects. Right click on the resource item, and click **Delete**.
3. Go to the **Packages** view, and delete the **com.dell.plugin.asm** package along with the contents of the package.
This action also deletes the workflow and actions.
4. Login to the vRO server shell using SSH or console.
5. Change the working directory to `/var/lib/vco/app-server/plugins/`. Remove the **o11nplugin-asmvroplugin.dar** file.
6. Restart the vRO configurator and the application server. To restart the configurator server, run the command `/etc/init.d/vco-configurator restart` in the shell with root privilege. To restart the vRO application server, either run the command `/etc/init.d/vco-server restart` in the shell or use the vRO configurator. To restart the vRO application server through the configurator, login to the vRO configurator web console. Click **Startup** Options on the left pane. When the **Startup option** page is displayed on the right panel, click the **Restart Service** link.
The ASM vRO Integration is uninstalled.
7. Start the vRO console to verify if the integration is uninstalled.